In the Claims:

Please amend claims 1 and 10 as shown below.

1. (Currently amended) A distributed data system, comprising:

a plurality of nodes coupled together, wherein each one of the plurality of nodes is coupled to at least one other one of the plurality of nodes for communicating data between the nodes, wherein the plurality of nodes comprises:

at least one in-process node comprising:

an in-process client; and

~~and~~ a distributed data manager, wherein the in-process client and the distributed data manager for the in-process node are configured to execute within the same computer process on the in-process node, and wherein the distributed data manager for the in-process node is configured to communicate data with the in-process client in a non-serialized format and communicate data with other ones of the plurality of nodes in a serialized format; and

at least one out-of-process node comprising an out-of-process client, wherein the out-of-process client is configured to execute within a different process than any distributed data manager, and wherein the out-of-process client is configured to communicate data with other processes or other ones of the nodes in a serialized format.

2. (Original)   The system as recited in claim 1, wherein the in-process client is configured to request data from the distributed data manager for the in-process node, and wherein, in response to the client's request, the distributed data manager for the in-process node is configured to return a reference to an object for the data without serializing the data.

3. (Original)   The system as recited in claim 2, wherein the distributed data manager for the in-process node is configured to receive a request for data from another node, serialize the requested data and send the serialized data to the requesting node.

4. (Original)   The system as recited in claim 1, wherein the distributed data manager for the in-process node is configured to receive serialized data from another node, de-serialize the data and store the de-serialized data as an object.

5. (Original)   The system as recited in claim 1, wherein said in-process client is configured to send an object reference for the data to the distributed data manager for the in-process node to store data in the in-process node.

6. (Original)   The system as recited in claim 1, wherein all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node.

7. (Original)   The system as recited in claim 1, further comprising a distributed data manager for the out-of-process node, wherein the distributed data manager for the out-of-process node is configured to communicate data with other processes or other ones of the plurality of nodes in a serialized format.

8. (Original)   The system as recited in claim 7, wherein the out-of-process client is configured to execute in a separate process and communicate data with the distributed data manager for the out-of-process node in a serialized format.

9. (Original)   The system as recited in claim 7, wherein the out-of-process client is configured to request data from the distributed data manager for the out-of-process node, and wherein, in response to the client's request, the distributed data manager for the out-of-process node is configured to send the requested data to the out-of-process client in a serialized format.

10. (Currently amended) The system as recited in claim 7, wherein said out-of-process client is configured to send serialized data to the distributed data manager for the out-of-process node to store data~~to the distributed data manager for the out-of-process node~~, wherein the distributed data manager for the out-of-process node is configured to store the data in its serialized format.

11. (Original) The system as recited in claim 7, wherein the distributed data manager for the out-of-process node is configured to receive serialized data from another node, and store the data in its serialized format.

12. (Original) The system as recited in claim 7, wherein all data store operations performed by the distributed data manager for the out-of-process node store data in a serialized format in a data store of the out-of-process node.

13. (Original) The system as recited in claim 7, wherein the distributed data manager for the out-of-process node is configured to replicate data to one or more other ones of the plurality of nodes.

14. (Original) The system as recited in claim 7, wherein the distributed data manager for the out-of-process node is comprised within an application server, and wherein the out-of-process client is a web server coupled to the application server.

15. (Original) The system as recited in claim 1, wherein the distributed data manager for the in-process node is configured to replicate data stored in the in-process node to one or more other ones of the plurality of nodes.

16. (Original) The system as recited in claim 1, wherein the distributed data manager for the in-process node is comprised within an application server, and wherein the in-process client is a web server coupled to the application server.

17. (Original) A method, comprising:

an in-process client requesting data from a distributed data manager for an in-process node of a distributed data system, wherein the in-process client and the distributed data manager for the in-process node execute within the same process on the in-process node;

if the requested data is present in a data store managed by the distributed data manager for the in-process node:

the distributed data manager for the in-process node returning the requested data to the in-process client as an object without serializing the data;

if the requested data is not present in the data store managed by the distributed data manager for the in-process node:

the distributed data manager for the in-process node retrieving the requested data in a serialized format from another node of the distributed data system;

the distributed data manager for the in-process node de-serializing the data retrieved from another node into an object; and

the distributed data manager for the in-process node returning the requested data to the in-process client as the de-serialized object;

an out-of-process client requesting data from a node within the distributed data system; and

the out-of-process client receiving the requested data in a serialized format.

18. (Original) The method as recited in claim 17, further comprising:

the distributed data manager for the in-process node receiving a request for data from another node;

the distributed data manager for the in-process node serializing the requested data; and

the distributed data manager for the in-process node sending the serialized data to the requesting node.

19. (Original) The method as recited in claim 17, further comprising the distributed data manager for the in-process node receiving serialized data from another node, de-serializing the data and storing the de-serialized data as an object.

20. (Original) The method as recited in claim 17, further comprising said in-process client sending an object reference for the data to the distributed data manager for the in-process node for storing data in the in-process node.

21. (Original) The method as recited in claim 17, wherein all data store operations performed by the distributed data manager in the in-process node comprise storing data in a non-serialized object format in a data store of the in-process node.

22. (Original) The method as recited in claim 17, wherein the node within the distributed data system from which the out-of-process client requests data is configured

as an out-of-process node comprising a distributed data manager, wherein the distributed data manager for the out-of-process node is configured to communicate data with other processes or other nodes of the distributed data system in a serialized format.

23. (Original) The method as recited in claim 22, further comprising the out-of-process client executing in a separate process and communicating data with the distributed data manager for the out-of-process node in a serialized format.

24. (Original) The method as recited in claim 22, further comprising:

the out-of-process client requesting data from the distributed data manager for the out-of-process node; and

the distributed data manager for the out-of-process node sending the requested data to the out-of-process client in a serialized format in response to the client's request.

25. (Original) The system as recited in claim 22, further comprising said out-of-process client storing data in the out-of-process node by sending serialized data to the distributed data manager for the out-of-process node, wherein the distributed data manager for the out-of-process node is configured to store the data in its serialized format.

26. (Original) The method as recited in claim 22, further comprising the distributed data manager for the out-of-process node receiving serialized data from another node, and storing the data in its serialized format.

27. (Original) The method as recited in claim 22, wherein all data store operations performed by the distributed data manager for the out-of-process node comprise storing data in a serialized format in a data store of the out-of-process node.

28. (Original) The method as recited in claim 22, further comprising the distributed data manager for the out-of-process node replicating data stored in the out-of-process node to other nodes of the distributed data system in a serialized format.

29. (Original) The method as recited in claim 22, wherein the distributed data manager for the out-of-process node is comprised within an application server, and wherein the out-of-process client is a web server coupled to the application server.

30. (Original) The method as recited in claim 17, further comprising the distributed data manager for the in-process node replicating data stored in the in-process node to other nodes of the distributed data system in a serialized format.

31. (Original) The method as recited in claim 17, wherein the distributed data manager for the in-process node is comprised within an application server, and wherein the in-process client is a web server coupled to the application server.

32. (Original) A method, comprising:

an out-of-process client requesting data from a distributed data manager for an out-of-process node of a distributed data system, wherein the out-of-process client and the distributed data manager for the out-of-process node execute in two distinct processes;

if the requested data is present in a data store managed by the distributed data manager for the out-of-process node:

the distributed data manager for the out-of-process node returning the requested data to the out-of-process client as a serialized object;

if the requested data is not present in the data store managed by the distributed data manager for the out-of-process node:

the distributed data manager for the out-of-process node retrieving the requested data in a serialized format from another node of the distributed data system; and

the distributed data manager for the out-of-process node returning the requested data in a serialized format to the out-of-process client;

an in-process client requesting data from a distributed data manager for an in-process node of the distributed data system, wherein the in-process client and the distributed data manager for the in-process node execute within the same process on the in-process node; and

the in-process client receiving the requested data in de-serialized format.

33. (Original) The method as recited in claim 32, further comprising the distributed data manager for the out-of-process node storing the data retrieved from another node in a serialized format in the data store managed by the distributed data manager for the out-of-process node.